CSE535 Project3: Evaluation of IR Models

Mansi Wagh
UBIT name: mansiwag
Department of Computer Science
University at Buffalo
Buffalo, NY 14226
mansiwag@buffalo.edu

Abstract

This project was implemented to evaluate various Information Retrieval Models and interpret the results of each model. Models like Language model, BM25 Model and Divergence from randomness (DFR) Models were evaluated based on their similarity and by tweaking their parameters to get the best Mean Average Precision(MAP) values from the models against the queries. In order to increase the relevance of document terms against the given queries various strategies were applied on each model to improve the MAP value and hence enhance their performance.

1.Introduction

Information retrieval (IR) is concerned with representing, searching, and manipulating large collections of electronic text data. IR is the discipline that deals with retrieval of unstructured data or partially structured data, especially textual documents, in response to a set of query or topic statement(s), which may itself be unstructured. The typical interaction between a user and an IR system can be modeled as the user submitting a query to the system; the system returns a ranked list of relevant documents, with the most relevant at top of the list. The need for effective methods of automated IR has grown in importance because of the tremendous explosion in the amount of text documents and growing number of document sources on the Internet.

In this project, we have studied and implemented Language model, Best Matching(BM25) model and Divergence from randomness Model(DFR) to retrieve relevant documents for the sample queries.

2. Dataset Description:

We have used Twitter data saved in json format, training_tweet.json for experimenting with our Models. The training_tweet.json file contains approximately 3,500 tweets with some fields extracted from raw data. Our dataset includes three languages- English (text_en), German (text_de) and Russian (text_ru).

3. Input Queries:

The IR models were tested against 15 different queries in English, German and Russian language and corresponding manually judged relevance score.

4. IR Models Evaluations:

We have implemented following 3 models in this project:

4.1 Language Model:

The goal of probabilistic language model is to calculate the probability of a sentence of sequence of words that can be used to find the probability of the next word in the sequence. A model that computes either of these is called a Language Model.

We have implemented Language Model in Solr by creating a Language model core and including the LMJelinekMercerSimilarityFactory in schema of our model.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Solr managed schema - automatically generated - DO NOT EDIT -->
<schema name="default-config" version="1.6">
  <uniqueKey>id</uniqueKey>
  <similarity class="solr.LMJelinekMercerSimilarityFactory">
  <float name="lambda">0.9</float>
  </similarity>
```

LMJelinekMercerSimilarityFactory has a smoothing parameter lambda. The optimal value of lambda depends on both the collection and the query. The optimal value is around 0.1 for title queries and 0.7 for long queries. A high value of lambda tends to retrieve documents containing all query words while a low value of lambda is disjunctive and suitable for long queries. So choosing a correct value of lambda is very important for good performance of model.

Following are MAP values against various lambda values

Lamba	0.99	0.2	0.6	0.12	0.45
MAP	0.5947	0.6732	0.6719	0.6750	0.6749

Table 1: MAP values for Language Model against various lambda values

4.2 Best Matching(BM25) Model:

BM25(standing for Best Match 25) is a popular ranking function that can quantify the importance of the presence of each word/token/term for a given document. It is simply an improvement over TF-IDF (Term Frequency Inverse Document Frequency).

We have implemented BM25 Model in Solr by creating a BM25 model core and including the BM25SimilarityFactory in schema of our model

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Solr managed schema - automatically generated - DO NOT EDIT -->
<schema name="default-config" version="1.6">
  <uniqueKey>id</uniqueKey>
<similarity class="solr.BM25SimilarityFactory">
<float name="k1">1</float>
<float name="b">0.5</float>
</similarity>
```

BM25 Model has 2 parameters k1 and b. K1 controls non-linear term frequency normalization (saturation) and its default value is 1.2. b parameter controls to what degree document length normalizes tf values and its default value is 0.75. A higher k1 means that the score for each term can continue to go up by relatively more for more instances of that term. A value of 0 for k1 would mean that everything except IDF(q_i) would cancel out. Following is the table showing various MAP values with different pairs of k1 and b values.

K1	1.2	1	0	0.8	2	0	2.8
B	0.7	0.5	0.3	0.8	0	0.9	0.2
MAP	0.6658	0.6605	0.6758	0.6670	0.6612	0.6758	0.6485

Table 2: BM25 Model MAP values against different k1 & b values

From above table we infer that, we get better MAP values for lower k1 value and b values in the range 0.3-0.9.

4.3 Divergence from Randomness (DFR) Model:

DFR is a probabilistic model used to test the amount of information carried in the documents. DFR models set up by instantiating the three main components of the framework: first selecting a basic randomness model, then applying the first normalization and at last normalizing the term frequencies

We have implemented DFR Model in Solr by creating a DFR model core and including the DFRSimilarityFactory in schema of our model.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- Solr managed schema - automatically generated - DO NOT EDIT -->
<schema name="default-config" version="1.6">
  <uniqueKey>id</uniqueKey>
  <similarity class="solr.DFRSimilarityFactory">
  <str name="basicModel">G</str>
  <str name="afterEffect">B</str>
  <str name="normalization">H2</str>
  <float name="c">7</float>
```

Here, we have chosen Basic Model “G”, Bernoulli first normalization ”B” and “H2” second normalization as components of DFR. Following table shows the MAP values with varying second normalization values(c).

C values	7	5	0.5	10	1	9
MAP	0.6761	0.6786	0.6534	0.6786	0.6718	0.6761

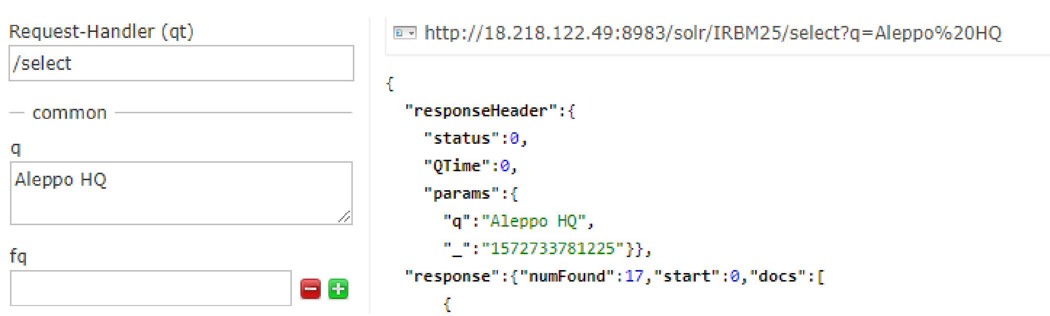
Table 3: MAP values against 2nd normalization c values

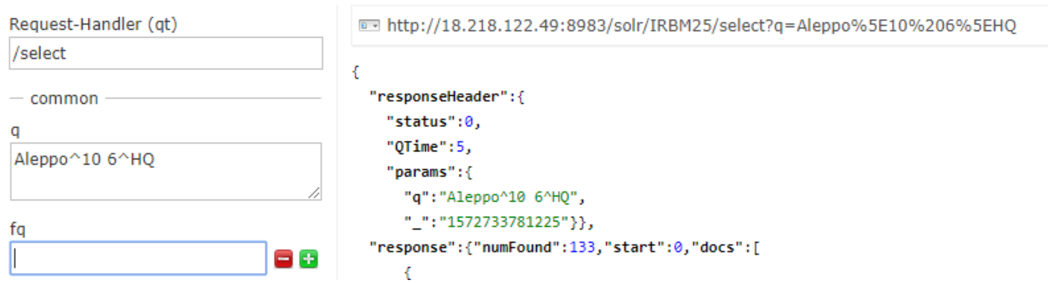
5. Experimental Results of Techniques tried to improve MAP values of each model:

Users always want relevant documents retrieved for the query they run. Giving higher relevance to a set of documents over others is called boosting. Our MAP value would increase with high relevance documents.

5.1.Boosting query terms:

We can retrieve more relevant documents by boosting query terms. For example in Solr against our sample test query say 010 Aleppo HQ we get only 17 documents with comparatively low relevance. But on boosting query terms manually, we retrieve 133 documents with higher relevance.





5.2 Boosting with eDismax Query Parser:

When the queries are parsed using eDismax Query Parser, the MAP values are significantly increased for all the 3 models. We make changes in solrconfig.xml to use eDismax query parser for sample queries.

```
<lst name="defaults">
  <str name="echoParams">explicit</str>
  <int name="rows">10</int>
      <str name="defType">edismax</str>
  <str name="qf">text_en text_de text_ru</str>
  <!-- Default search field
```

Following table compares MAP values for all models before and after putting edismax query parser.

MAP Values	Language Model	BM25 Model	DFR Model
Before eDismax	0.2345	0.2685	0.2434
After eDismax	0.6534	0.6271	0.5783

Table 4: MAP values of all models before and after edismax query parser was used.

5.3 Using Various Filter Factory:

Following filters were used for English language.

```
<filter class="solr.SynonymGraphFilterFactory" expand="true" ignoreCase="true" synonyms="synonyms.txt"/>
<filter class="solr.StopFilterFactory" words="lang/stopwords_en.txt" ignoreCase="true"/>
<filter class="solr.LowerCaseFilterFactory"/>
<filter class="solr.EnglishPossessiveFilterFactory"/>
<filter class="solr.EnglishMinimalStemFilterFactory"/>
<filter class="solr.KeywordMarkerFilterFactory" protected="protwords.txt"/>
<filter class="solr.PorterStemFilterFactory"/>
```

We used following filters for German language in schema of our model

```
<filter class="solr.LowerCaseFilterFactory"/>
<filter class="solr.StopFilterFactory" format="snowball" words="lang/stopwords_de.txt" ignoreCase="true"/>
<filter class="solr.GermanNormalizationFilterFactory"/>
<filter class="solr.GermanLightStemFilterFactory"/>
```

Following filters were used for Russian language:

```
<filter class="solr.LowerCaseFilterFactory"/>
<filter class="solr.StopFilterFactory" format="snowball" words="lang/stopwords_ru.txt" ignoreCase="true"/>
<filter class="solr.SnowballPorterFilterFactory" language="Russian"/>
```

6. Results:

Following are the evaluations for each model based on the techniques applied to improve the MAP values.

	Default MAP values	After using eDismax query Parser	After tweaking parameters	Best MAP Value
Language Model	0.2345	0.6534	0.6605	0.6750
BM25 Model	0.2685	0.6271	0.6721	0.6758
DFR Model	0.2434	0.5783	0.6634	0.6786

6.1 Language Model:

Best MAP value obtained for Language Model in TREC evaluation by executing all 15 sample queries is 0.6750 i.e 67.5% . It is shown in below screenshot

```
runid                   all     LM
num_q                   all     15
num_ret                 all     260
num_rel                 all     225
num_rel_ret             all     116
map                     all     0.6750
gm_map                  all     0.3102
Rprec                   all     0.6830
bpref                   all     0.6926
recip_rank              all     0.9333
```

6.2 BM25 Model:

Best MAP value obtained for BM25 Model in TREC evaluation by executing all 15 sample queries is 0.6758 i.e 67.58% . It is shown in below screenshot

```
runid                    all     BM25
num_q                    all     15
num_ret                  all     260
num_rel                  all     225
num_rel_ret              all     113
map                      all     0.6758
gm_map                   all     0.3073
Rprec                    all     0.6942
bpref                    all     0.6993
recip_rank               all     0.9333
```

3.DFR Model:

Best MAP value obtained for DFR Model in TREC evaluation by executing all 15 sample queries is 0.6786 i.e 67.86% . It is shown in below screenshot

```
runid                    all     DFR
num_q                    all     15
num_ret                  all     260
num_rel                  all     225
num_rel_ret              all     118
map                      all     0.6786
gm_map                   all     0.3098
Rprec                    all     0.6804
bpref                    all     0.7053
recip_rank               all     0.9333
```

7.Conclusion:

Hence, we have successfully implemented the Language Model, BM25 Model and Divergence from Randomness Model(DFR) and obtained optimal MAP values for each of them by their TREC evaluation.

8.References:

1.lucene.apache.org/solr/guide/7_5/otherschemaelements.html#OtherSchemaElements-Similarity.

2. https://cwiki.apache.org/confluence/display/solr/SchemaXml#Similarity

3.lucene.apache.org/solr/7_0_0/solrcore/org/apache/solr/search/similarities/LMJelinekMercerSimilarityFactory.html

4**.** pdfs.semanticscholar.org/17e4/deb8193dc274255cc751309085bf0af02885.pdf

5.https://towardsdatascience.com/learning-nlp-language-models-with-real-data-cdff04c51c25